
newslynx-sc-google-analytics

Release 0.0.1

Sep 27, 2017

Contents

1	Installation	1
1.1	Production	1
1.2	Development	1
2	Tests	3
3	Documentation	5
4	Continuous Integration	7
5	Contributing	9
6	What's in this module ?	11
7	Contents	13
7.1	Sous Chefs	13

CHAPTER 1

Installation

Production

To install `newslynx-sc-google-analytics` for an active installation of `newslynx-core`, run the following command:

```
$ newslynx sc-install https://github.com/newslynx/newslynx-sc-google-analytics.git
```

To add `newslynx-sc-google-analytics` all orgnaizations, run:

```
$ newslynx sc-sync
```

Development

If you want to modify / add Sous Chefs to `newslynx-sc-google-analytics`, do the following:

NOTE Will install a fresh version of `newslynx` via `pip`.

```
$ git clone https://github.com/newslynx/newslynx-sc-google-analytics.git
$ cd newslynx-sc-google-analytics
$ pip install --editable .
```

You should now be able to run `newslynx-sc-google-analytics`'s Sous Chefs in development mode

```
% newslynx sc-run newslynx_sc_google_analytics/say_my_name.yaml --myname='Brian_
↪Abelson'
```


CHAPTER 2

Tests

Requires nose

```
$ make all_tests
```


CHAPTER 3

Documentation

Documentation for `newslynx-sc-google-analytics` is hosted on [Read The Docs](#).

It's generated via the following steps

- converts this file (`README.md`) into a ReStructured Text file, saving it to `docs/index.rst`
- runs `newslynx sc-docs newslynx_sc_google_analytics -f rst` to generate documentation for all the Sous Chefs in `newslynx-sc-google-analytics` and saves the output to `docs/sous-chefs.rst`
- Builds Sphinx Documentation from these files.

CHAPTER 4

Continuous Integration

Builds for `newslinx-sc-google-analytics` can be found on [Travis](#)

CHAPTER 5

Contributing

See the [contributing guidelines](#).

What's in this module ?

- `README.md`
 - This file
- `VERSION`
 - `newslynx-sc-google-analytics`'s source-of-truth version.
- `requirements.txt`
 - `newslynx-sc-google-analytics`'s python dependencies.
- `MANIFEST.in`
 - Specifications for which files to include in the PyPI distribution.
 - See the docs on this [here](#).
- `setup.py`
 - Specification's for building `newslynx-sc-google-analytics`'s PyPI distribution.
- `.travis.yml`
 - Configurations for Travis Continuous Integration
 - You must activate this project on [travis-ci.org](#) for this to run on subsequent updates.
- `Makefile`
 - Helpers for managing `newslynx-sc-google-analytics`.
 - Includes:
 - * `make clean`:
 - Cleans out cruft from this directory.
 - * `make install`:
 - Installs `newslynx-sc-google-analytics`. Assumes that you're in a virtual environment.
 - * `make all_tests`:

- Runs the tests.
- * make readme
 - Converts this file to `.rst`, including a table of contents, and saves it to `docs/index.rst`
- * make sous_chef_docs
 - Programmatically generates `Sous Chef` documentation by running `newslynx sc-docs newslynx_sc_google_analytics/ --format=rst > docs/sous-chefs.rst`.
- * make all_docs:
 - Builds the sphinx docs for `newslynx-sc-google-analytics` by running the above two commands.
- * make view_docs
 - Serves documentation at `localhost:8000`
- * make register:
 - Registers `newslynx-sc-google-analytics` on `PyPI`.
- * make distribute:
 - Publishes a new version of `newslynx-sc-google-analytics` to `PyPI`.
- `CONTRIBUTING.md`
- `newslynx_sc_google_analytics`
 - `newslynx-sc-google-analytics`'s source code and `Sous Chef` configuration files.
- `docs`
 - Sphinx documentation for `newslynx-sc-google-analytics`
- `tests`
 - nose tests for `newslynx-sc-google-analytics`

Sous Chefs

newslynx-sc-google-analytics provides access to the following Sous Chefs

Google Analytics Pageviews by Device Type for content Items

- Computes the total pageviews for content items by device type.
- This Sous Chef runs the python module `newslynx_sc_google_analytics.ContentDeviceSummaries`.
- API Slug: `google-analytics-to-content-device-summaries`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↪content_device_summaries.yaml --passthrough **options
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↪content_device_summaries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_google_analytics/google_analytics_to_
↳content_device_summaries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-device-summaries_
↳**options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-device-summaries -
↳-data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `google-analytics-to-content-device-summaries` and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_device_
↳summaries.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_device_
↳summaries.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `google-analytics-to-content-device-summaries` also accepts the following

- `days`
 - The number of days past a content item's creation date after which we will stop computing these counts.
 - Should be rendered with a number form.
 - Accepts inputs of type:
 - * `numeric`
 - Defaults to 30
- `content_item_types`

- The content item types to calculate share counts for.
- Should be rendered with a `text` form.
- Choose from:
 - * `video`
 - * `article`
 - * `slideshow`
 - * `interactive`
 - * `podcast`
 - * `all`
- Accepts inputs of type:
 - * `string`
- Defaults to `all`

Metrics

`google-analytics-to-content-device-summaries` generates the following Metrics

- `ga_pageviews_mobile`
 - **Display name:** Mobile Pageviews
 - **Type:** `count`
 - **Content Levels:**
 - * `summary`
 - * `comparison`
 - **Org Levels:**
 - * `summary`
- `ga_pageviews_tablet`
 - **Display name:** Tablet Pageviews
 - **Type:** `count`
 - **Content Levels:**
 - * `summary`
 - * `comparison`
 - **Org Levels:**
 - * `summary`
- `ga_pageviews_desktop`
 - **Display name:** Desktop Pageviews
 - **Type:** `count`
 - **Content Levels:**
 - * `summary`

- * comparison
- Org Levels:
- * summary

Google Analytics Domain Facets For Content Items

- Creates faceted metrics for pageviews by referrer for content items.
- This Sous Chef runs the python module `newslynx_sc_google_analytics.ContentDomainFacets`.
- API Slug: `google-analytics-to-content-domain-facets`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↳content_domain_facets.yaml --passthrough **options
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↳content_domain_facets.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_google_analytics/google_analytics_to_
↳content_domain_facets.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-domain-facets_
↳**options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-domain-facets --
↳data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to `google-analytics-to-content-domain-facets` and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_domain_
↪ facets.yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_domain_
↪ facets.yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, `google-analytics-to-content-domain-facets` also accepts the following

- `days`
 - The number of days past a content item's creation date after which we will stop computing these counts.
 - Should be rendered with a number form.
 - Accepts inputs of type:
 - * `numeric`
 - Defaults to 30
- `max_facets`
 - The maximum number of facets to include. Will sort by pageviews.
 - Should be rendered with a number form.
 - Accepts inputs of type:
 - * `numeric`
 - Defaults to 20
- `content_item_types`
 - The content item types to calculate share counts for.
 - Should be rendered with a text form.
 - Choose from:
 - * `video`
 - * `article`
 - * `slideshow`
 - * `interactive`

- * podcast
- * all
- Accepts inputs of type:
 - * string
- Defaults to all

Metrics

google-analytics-to-content-domain-facets generates the following Metrics

- ga_pageviews_by_domain
 - Display name: Pageviews By Refering Domain
 - This is a **faceted** metric.
 - Type: count
 - Content Levels:
 - * summary
- ga_pageviews_by_article_referrer
 - Display name: Pageviews By Refering article.
 - This is a **faceted** metric.
 - Type: count
 - Content Levels:
 - * summary

Google Analytics Timeseries For Content Items

- Computes a timeseries of pageviews, entrances, and exits for content items.
- This Sous Chef runs the python module `newslynx_sc_google_analytics.ContentTimeseries`.
- API Slug: `google-analytics-to-content-timeseries`

Usage

Standalone

Run this Sous Chef via the api, passing in arbitrary runtime options, and stream it's output.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↪content_timeseries.yaml --passthrough **options
```

Run this Sous Chef via the api, and if applicable, send it's output to bulkload.

```
$ newslynx api sous-chefs cook -d=newslynx_sc_google_analytics/google_analytics_to_
↪content_timeseries.yaml **options
```

Do either of the above two, but pass in a recipe file

```
$ newslynx api sous-chefs cook -d=recipe.yaml
```

Recipes

Add this Sous Chef to your authenticated org

```
$ newslynx api sous-chefs create -d=newslynx_sc_google_analytics/google_analytics_to_
↳content_timeseries.yaml
```

Create a Recipe with this Sous Chef with command line options.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-timeseries_
↳**options
```

Alternatively pass in a recipe file.

```
$ newslynx api recipes create sous_chef=google-analytics-to-content-timeseries --
↳data=recipe.yaml
```

Save the outputted id of this recipe, and execute it via the API. **NOTE** This will place the recipe in a task queue.

```
$ newslynx api recipes cook id=<id>
```

Alternatively, run the Recipe, passing in arbitrary runtime options, and stream it's output: **NOTE** Will not execute the SousChef's load method.

```
$ newslynx api recipes cook id=<id> --passthrough **options
```

Development

Pass runtime options to google-analytics-to-content-timeseries and stream output. **NOTE** Will not execute the SousChef's load method.

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_timeseries.
↳yaml option=value1
```

Alternatively pass in a recipe file

```
$ newslynx sc-run newslynx_sc_google_analytics/google_analytics_to_content_timeseries.
↳yaml --recipe=recipe.yaml
```

Options

In addition to default recipe options, google-analytics-to-content-timeseries also accepts the following

- days
 - How many days back should we search each day
 - Should be rendered with a number form.
 - Accepts inputs of type:

- * numeric
- Defaults to 30
- content_item_types
 - The content item types to calculate share counts for.
 - Should be rendered with a `text` form.
 - Choose from:
 - * video
 - * article
 - * slideshow
 - * interactive
 - * podcast
 - * all
 - Accepts inputs of type:
 - * string
 - Defaults to `all`

Metrics

`google-analytics-to-content-timeseries` generates the following Metrics

- ga_pageviews
 - Display name: Pageviews
 - Type: count
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary
- ga_exits
 - Display name: Exits
 - Type: count
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:

- * timeseries
 - * summary
- ga_entrances
 - Display name: Entrances
 - Type: count
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary
- ga_total_time_on_page
 - Display name: Total Time on Page
 - Type: count
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary
- ga_avg_time_on_page
 - Display name: Average Time on Page
 - This is a **computed** metric with the formula:
 - * $\text{ROUND}(\{\text{ga_total_time_on_page}\} / \text{NULLIF}(\{\text{ga_pageviews}\}, 0), 2)$
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary
- ga_per_external
 - Display name: Percent External Traffic
 - This is a **computed** metric with the formula:

- * $\text{ROUND}(\{\text{ga_entrances}\} / \text{NULLIF}(\{\text{ga_pageviews}\}, 0), 2)$
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary
- ga_per_internal
 - Display name: Percent Internal Traffic
 - This is a **computed** metric with the formula:
 - * $\text{ROUND}(1 - (\{\text{ga_entrances}\} / \text{NULLIF}(\{\text{ga_pageviews}\}, 0)), 2)$
 - Content Levels:
 - * timeseries
 - * summary
 - * comparison
 - Org Levels:
 - * timeseries
 - * summary